بسم الله الرحمن الرحيم

Fuzzy Control Course

Lec 8

Introduction to population-based optimization

DR. M. Arafa

29/11/2016

Definition of Optimization

- In engineering, physical, and mathematical sciences, computational optimization, or simply optimization, means either minimization or maximization of a certain objective function.
- optimization is aimed to find the best (optimum) solution for any optimization problem.

Examples of Optimization Problems/ Applications

- Optimizing the parameters of the ANN model (weights and biases).
- Optimizing the parameters of the ANFIS model (premise and antecedent parameters).
- Tuning the parameters of the PID controller (get the best value for K_P, K_I and K_D).
- Getting the best Placement of Wi-Fi Access Point for Indoor Positioning system (IPS).
- Approximating experimental, mathematical functions with the least possible error.
- Optimizing the methods of Micro-array data analysis (Bioinformatics field).

Global and local optimum

- Optimum can be global or local. To illustrate, the single-variable function $f(x)$ (represents the objective function to be minimized) in Fig.1 has one global minimum and two local minimum ; the global minimum is the least among all local minimum.
- **Note that** every global optimum is a local optimum, but the reverse is not necessarily true.
- A unimodal function has a single local optimum which is itself the global optimum (as in Fig. 2).

Global and local optimum

- A multimodal function has more than one local optimum and one global optimum which is a local optimum with the least objective function value among all other local optimum.
- In an optimization problem, the ideal target is ofcourse the global optimum, and a 'good' optimization algorithm does not get trapped in any local optimum.

Global and local optimum

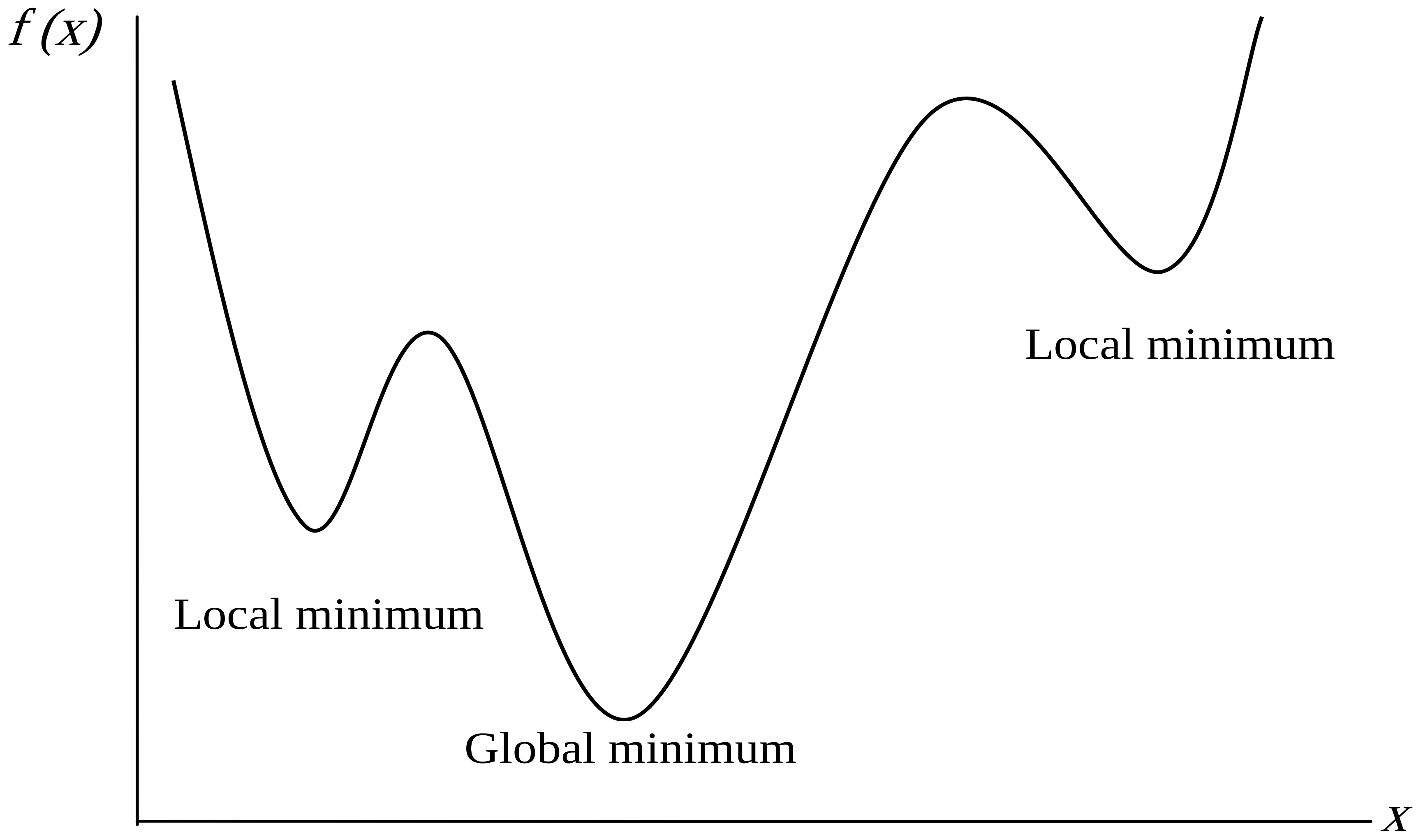


Fig. 1: f (X) is a multimodal function

Global and local optimum

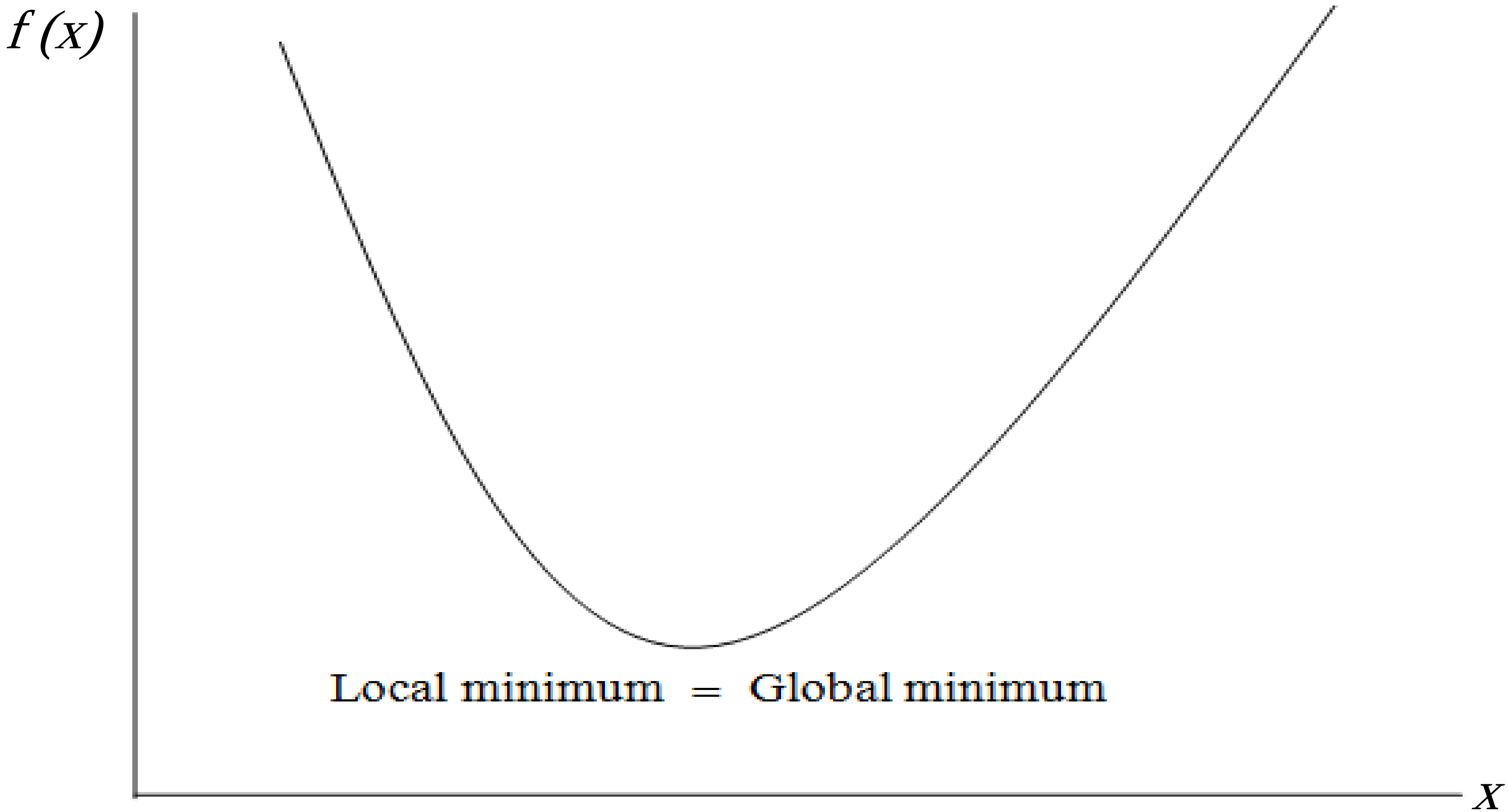


Fig. 2: $f\ (X)$ is a unimodal function

Basic Elements of Optimization Problem

(1) An objective function *f* which is the function to be optimized (minimized or maximized).

(2) The number of components or variables of the objective function that specifies the dimensionality of the optimization problem

If the objective function f is expressed in the form

$$f\ (x_1, x_2\ , \ldots\ldots., x_D)$$

Then $x_1, x_2\ , \ldots\ldots., x_D$ are the independent variables and D is the number of variables specifies the dimensionality of the problem.

The objective function can be written compactly as:

$$f\ (X), \quad X = [\, x_1, x_2\ , \ldots\ldots., x_D] \text{ is a 1xD vector}$$

Basic Elements of Optimization Problem

(3) A set of constraints forced on the required solution

Most problems constrain at least the search domains of the variables vector $X = [\, x_1, x_2 \,, \ldots\ldots, x_D \,]$.

- **Note that** the aim of an optimization method is to find the global optimum $X^* \subseteq R^D$ from the allowable search domains, such that $f(X^*)$ has the minimum value in the search domain.

Classification of Optimization Problems

Classification basis	**Types of optimization problems**	
Number D of variables x_d (Dimensionality)	**Univariate** ($D = 1$)	**Multivariate** ($D > 1$)
Linearity / nonlinearity of objective function	**Linear** (Objective function is linear in x_d)	**Nonlinear** (Objective function is nonlinear in x_d)
Constraints	**Unconstrained** (Only search ranges of x_d are constrained)	**Constrained** (Additional constraints are forced on x_d)

Classification basis	Types of optimization problems	
Number of optimum values	**Unimodal** (Objective function has one optimum only)	**Multimodal** (Objective function has more than one optimum)
Number of objectives	**Single-objective** (Single objective function is to be optimized)	**Multi-objective** (More than one objective function are to be optimized)
Separability of variables x_d	**Separable** If function $f\ (x_1, x_2\ ,\ \ldots\ldots.,\ x_D)$ can be divided to D functions in the following form: $f\ (x_1) + f(\ x_2) +\ \ldots\ldots f(x_D)$	**Non-separable** If function $f\ (x_1, x_2\ ,\ \ldots\ldots.,\ x_D)$ can not be divided to D functions in the following form: $f\ (x_1) + f(\ x_2) +\ \ldots\ldots f(x_D)$

Evolutionary Optimization Algorithms

(Population-based Optimization Algorithms)

Evolutionary Optimization Algorithms

- Evolutionary optimization algorithms are population-based algorithms of candidate solutions, not just one solution as in traditional methods.
- The basic characteristic of a population-based optimization method is that the iteration policy depends on a population.
- During the iterations, a population of constant size is maintained, and the group of solutions is improved progressively.
- Having a group of solutions 'working together' is the key to emulating the behavior of biological organisms in modern biology-inspired optimization approaches; examples are a flock of birds, a school of fish, and a colony of ants or honey bees.

Examples of Evolutionary Optimization Algorithms

- Genetic algorithm (GA)

  Holland, J. H. (1970)

- Ant colony optimization (ACO)

  Dorigo, M. (1992)

- Particle swarm optimization (PSO)

  Kennedy, J., and Eberhart, R. (1995)

- Differential evolution (DE)

  Storn, R., and Price, K.,(1997)

- Artificial bee colony (ABC)

  Karaboga (2005)

- Bat algorithm (BA)

  Yang (2010)

Exploration And Exploitation

- In the context of optimization, we have also two distinctive terms: **exploration** and **exploitation.**
  - **Exploration** means finding new solutions (or points) in the search domains which have not been evaluated before.

    In exploration, the variation of the population members from one iteration to another is large.
  - **Exploitation** means trying to improve the current found solutions by performing relatively small changes that lead to new solutions in the immediate neighborhood.

    In exploitation, the variation of the population members from one iteration to another is very small.

Basic elements Affect on Exploration And Exploitation

1) The population size (number of members in the population) affects on the exploration rate. Large size of the population, increase the rate of exploration.

2) The control parameters of the optimization algorithm affect on both of exploration and exploitation.

Note that:

Any optimization algorithm starts from large exploration rate and this allows the algorithm to cover large regions of search domains quickly. As iterations processes; the exploration rate is decreased, allowing to exploit the promising regions that that previously explored .

Testing An Optimization Algorithm

- In order to evaluate the efficiency and robustness for an optimization algorithm, standard complex mathematical functions with different characteristics called **benchmark functions** are used to test the optimization algorithm.
- After selecting a suitable set of benchmark functions, the algorithm is running over these functions for N independent of runs. Each run consists of determined No. of iterations.
- The results of the test show the No. of successful runs for each function. The run is considered successful if the algorithm reached to the required global optimum.

Testing An Optimization Algorithm

- There is a competition made every year between evolutionary optimization algorithms called **CEC** (Competition on Evolutionary Computation) to evaluate these algorithms determine the fittest optimization algorithm. These competitions use very complex benchmark functions with different characteristics.

(Ex: CEC 2005, CEC2006, CEC2007, …………., CEC2017.)

(http://www.ntu.edu.sg/home/epnsugan/index_files/cec-benchmarking.htm)

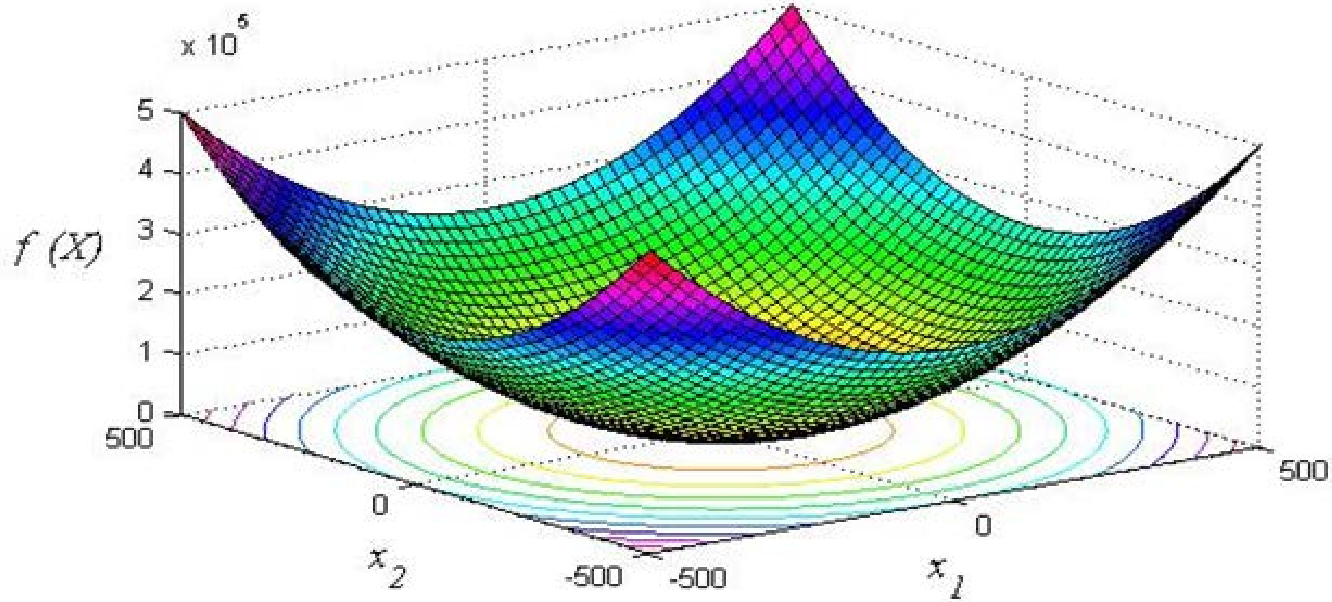
Common Benchmark Functions

Benchmark Functions		Search Range
Sphere Function	$f_1(x) = \sum_{i=1}^{i=D} x_i^2$	$[-100,100]^D$
Rosenbrock Function	$f_2(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-2.048,2.048]^D$
Ackley Function	$f_3(x) = 20 + e - 20\, e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}} - e^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)}$	$[-30,30]^D$
Griewank Function	$f_4(x) = 1 + \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-600,600]^D$
Rastrigin Function	$f_5(x) = \sum_{i=1}^{D} [10 + x_i^2 - 10\cos(2\pi x_i)]$	$[-5.12,5.12]^D$
Schwefel Function	$f_6(x) = 418.9829\, D - \sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500,500]^D$

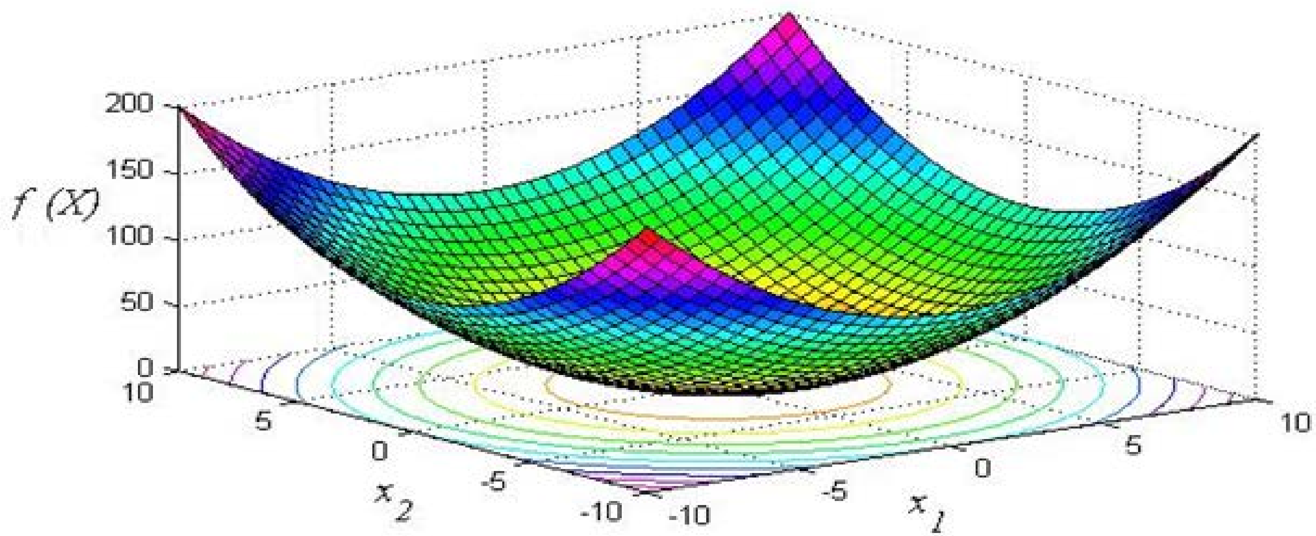
Functions Name	Functions properties
Sphere Function	- Unimodal - separable
Rosenbrock Function	- Unimodal ($D < 4$) - Multimodal ($D \geq 4$) - non-separable
Ackley Function	- Multimodal - non-separable
Griewank Function	- Multimodal - non-separable
Rastrigin Function	- Multimodal - separable
Schwefel Function	- Multimodal - separable

All of these functions are:

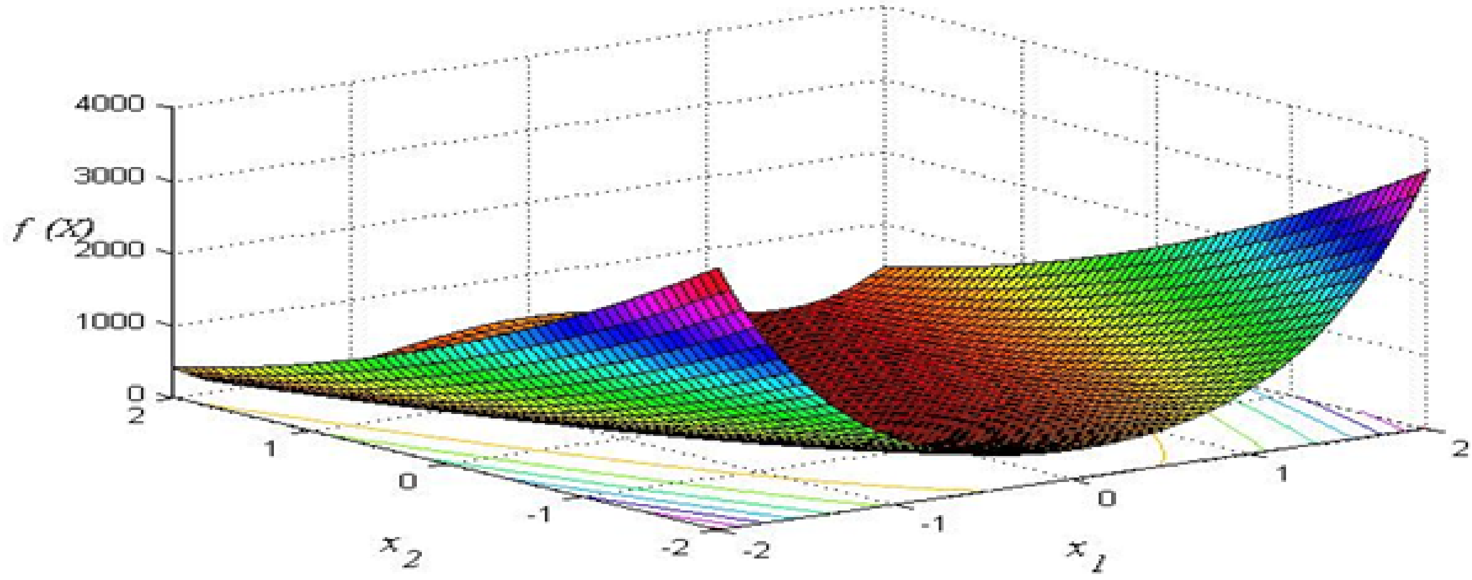
- **Single-objective**
- **Unconstrained**



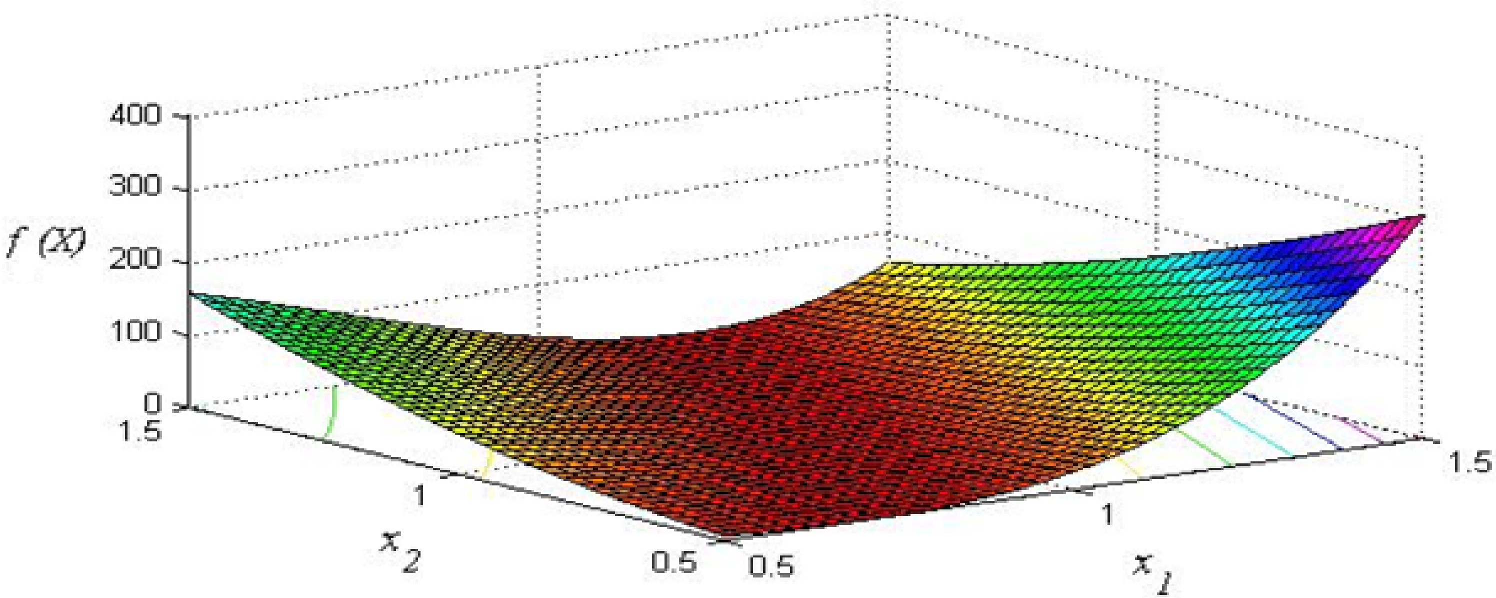
Sphere function, component range [-500,500]



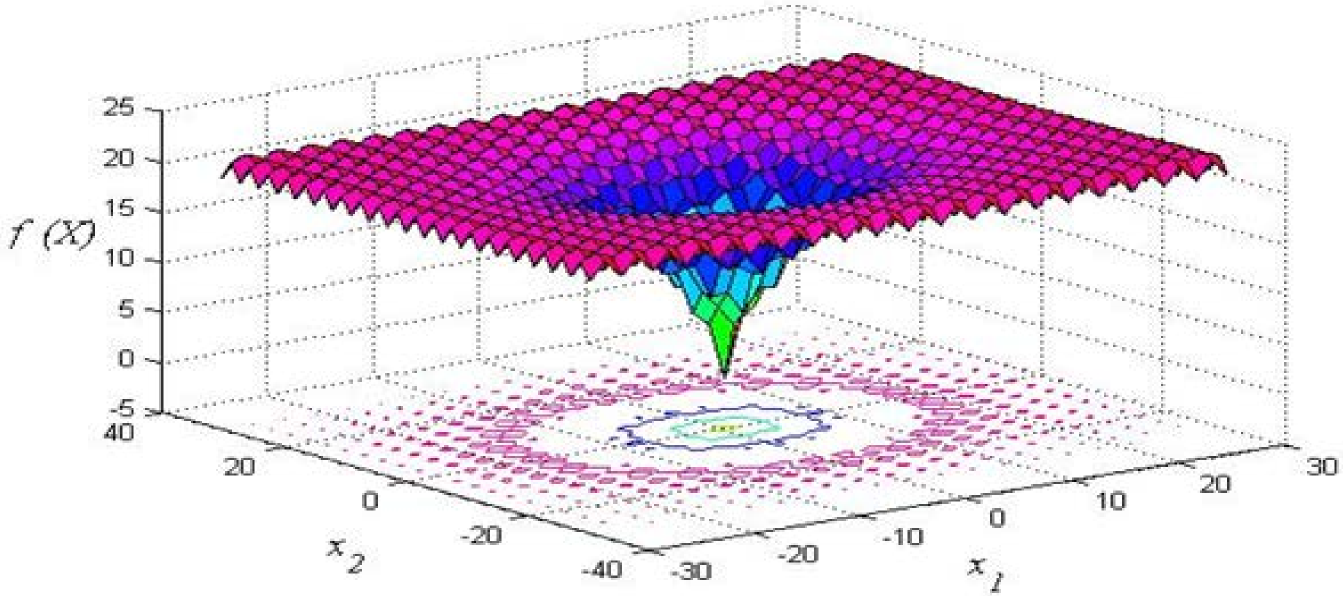
Sphere function, component range [-10,10]



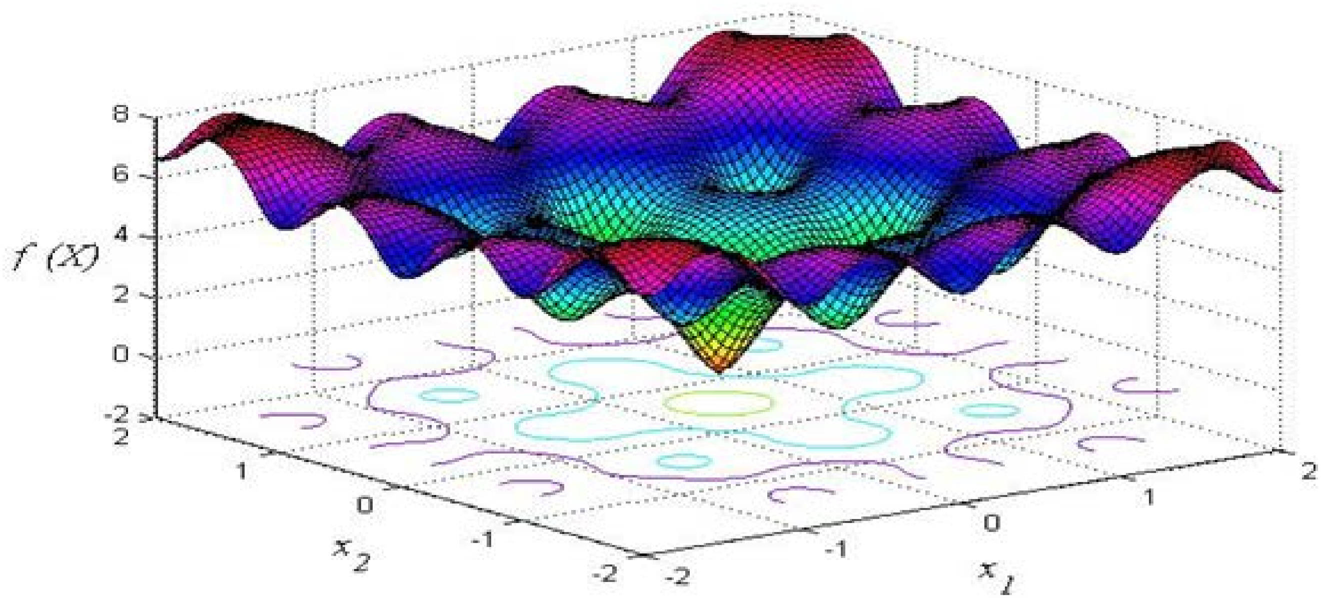
Rosenbrock function, component range [-2,2]



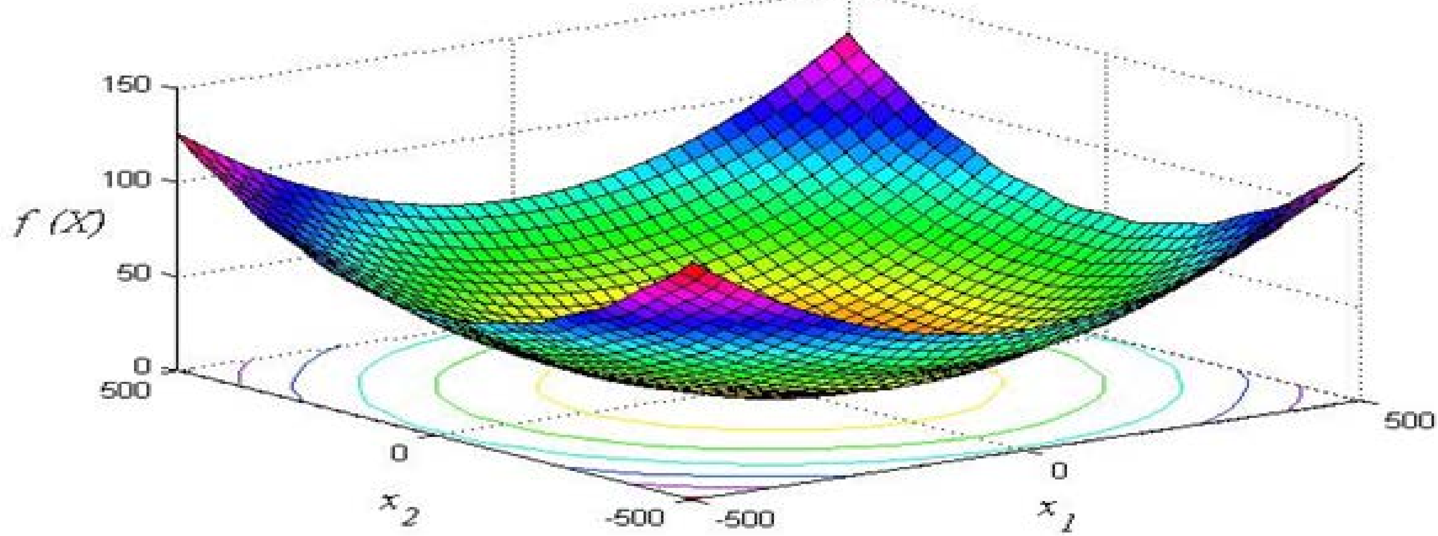
Rosenbrock function, component range [0.5,1.5]



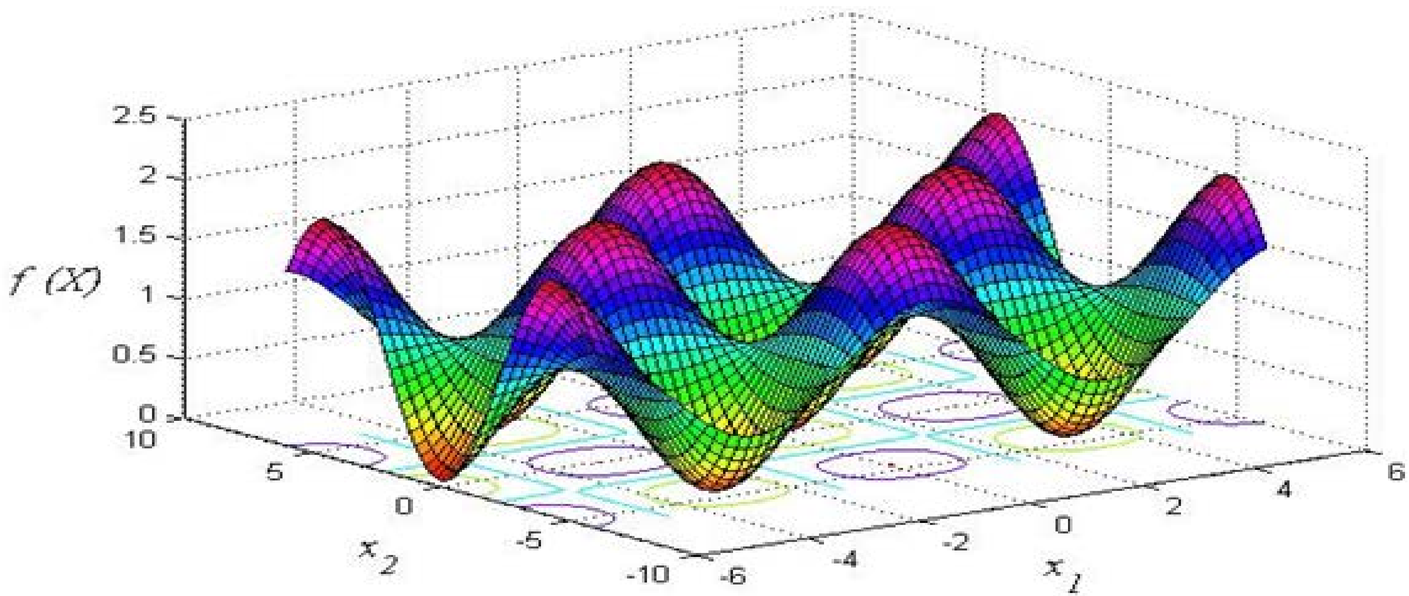
Ackley function, component range [-30,30]



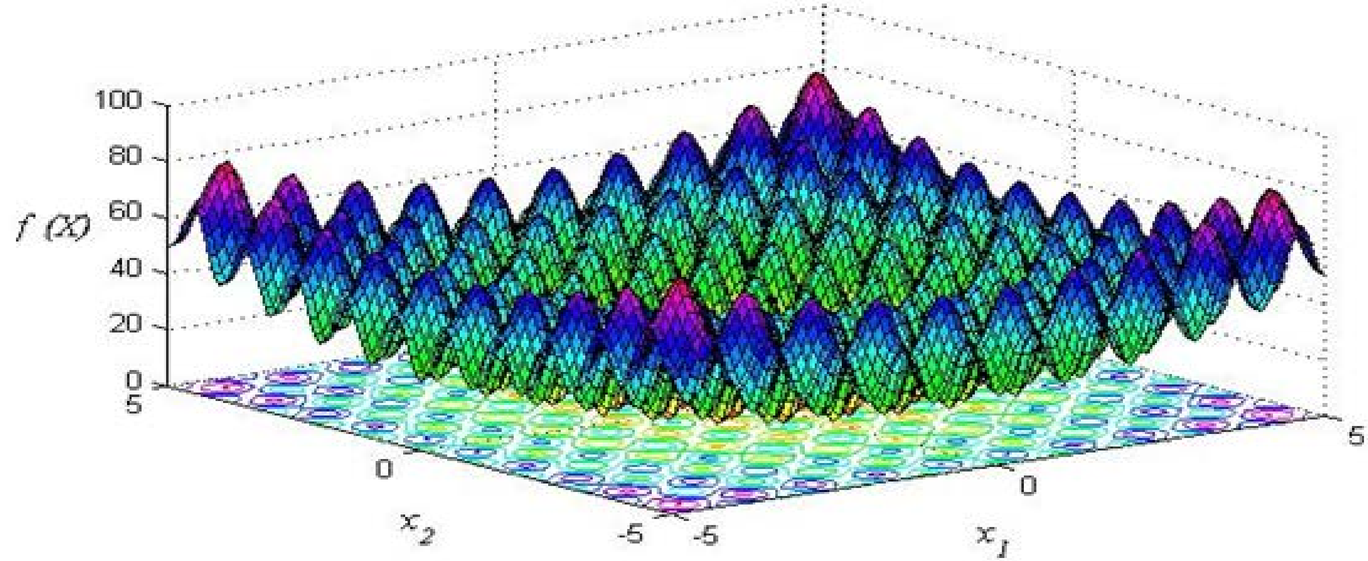
Ackley function, component range [-2,2]



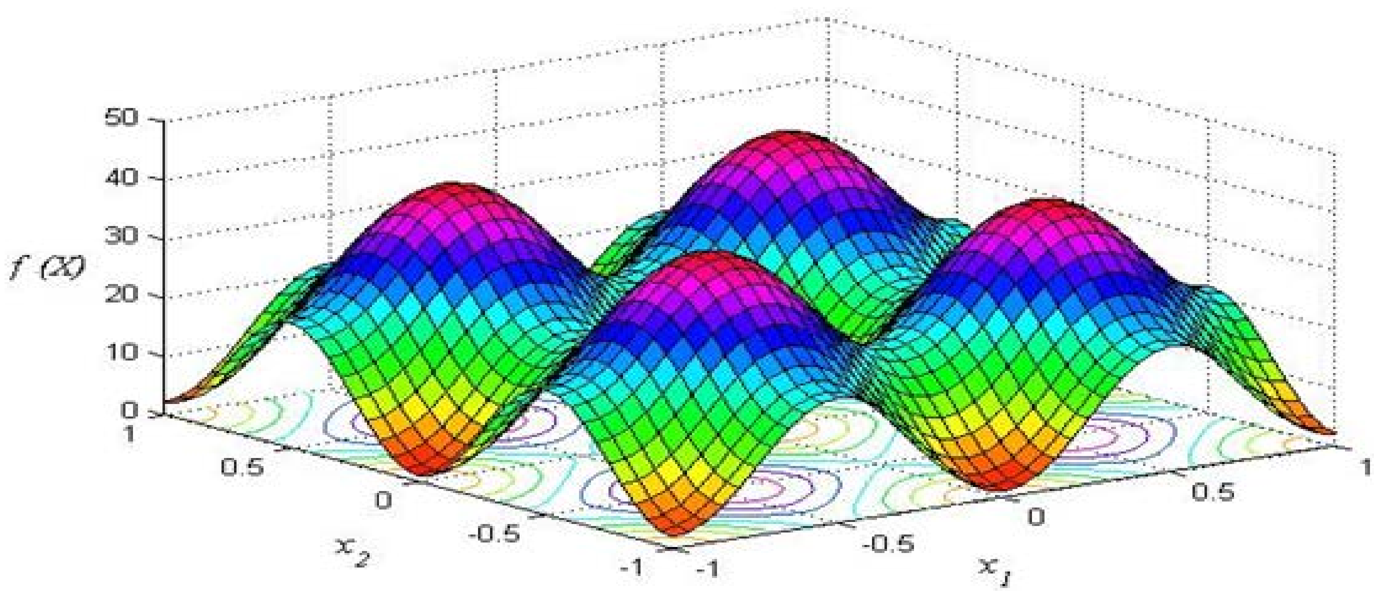
Griewank function, component range [-500,500]



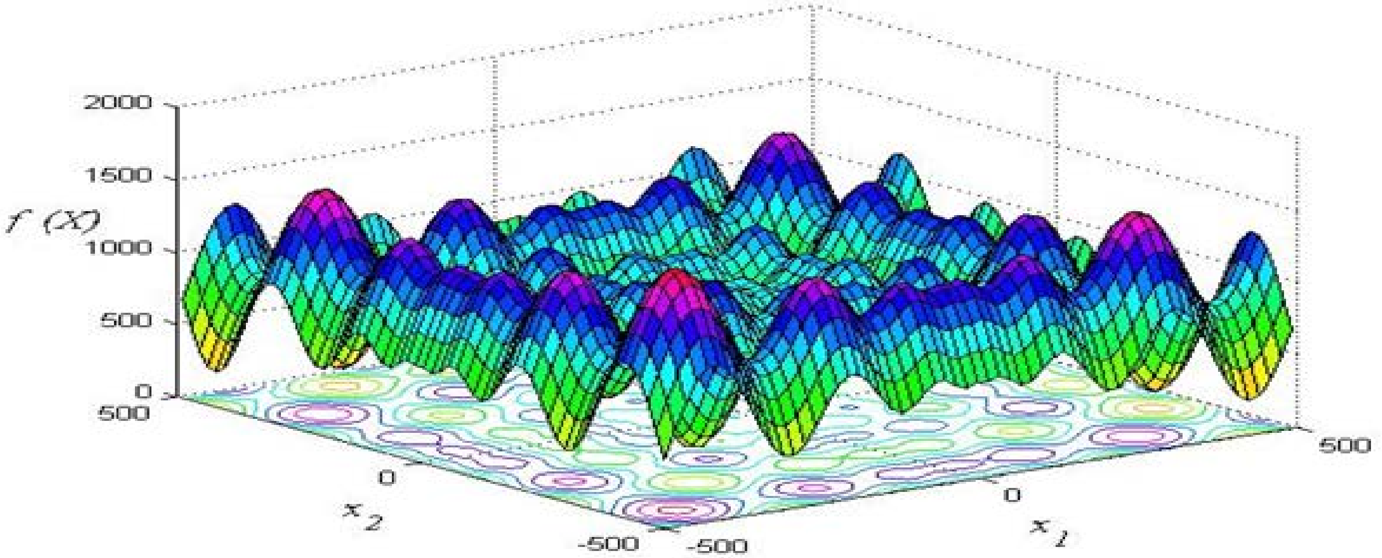
Griewank function, component range [-5,5]



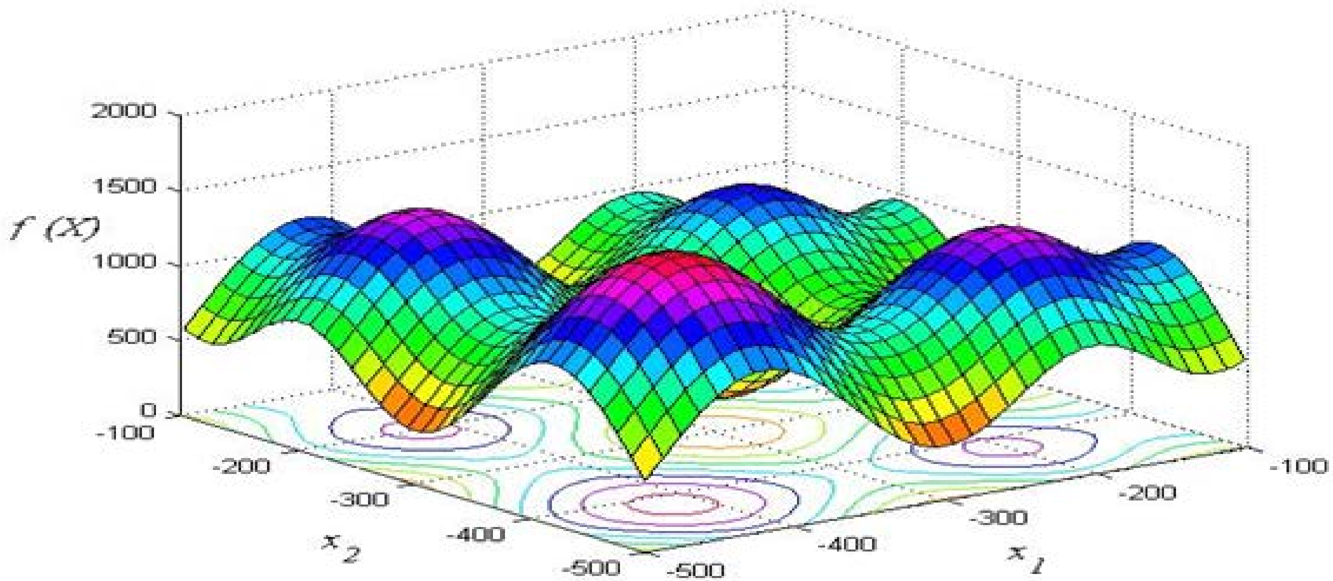
Rastrigin function, component range [-5,5]



Rastrigin function, component range [-1,1]



Schwefel function, component range [-500,500]



Schwefel function, component range [-500,-100]

MATLAB Code to Draw 3-D Plots of Benchmark Functions

```
% Example : Draw the 3-D map for Ackley function

>> x=[-5:0.1:5];

>>[x1 , x2] = meshgrid(x);

>> z = 20+exp(1) -20*exp(-0.2*sqrt(0.5*(x1.^2+x2.^2)))-
       exp(0.5*(cos(2*pi.*x1)+cos(2*pi.*x2)));

>> surfc(x1,x2,z)     %  Draw 3-D plot

>> colormap(hsv)     %  control the color map , for more colors, replace hsv with
% one of these words: jet – hot – cool – spring – summer – autumn – winter –
% gray – bone – copper – pink - lines
```